

## **JP2002084302**

Publication Title:

METHOD AND APPARATUS FOR COMMUNICATION BY NETWORK

Abstract:

Abstract of JP2002084302

**PROBLEM TO BE SOLVED:** To provide a method and an apparatus for communication by a virtual network, where independent protocol processing for each RS is realized, and a plurality of independent virtual networks can be realized. **SOLUTION:** An OS or a service layer individually recognizes a plurality of virtual data links, constituted on a physical data link, and the virtual data links are relayed independently.

Data supplied from the esp@cenet database - Worldwide

-----

Courtesy of <http://v3.espacenet.com>

(19) 日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号

特開2002-84302

(P2002-84302A)

(43) 公開日 平成14年3月22日 (2002.3.22)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	ページ* (参考)
H 0 4 L 12/46		G 0 6 F 9/46	3 4 0 A 5 B 0 8 9
12/28		13/00	3 5 3 C 5 B 0 9 8
G 0 6 F 9/46	3 4 0	H 0 4 L 11/00	3 1 0 C 5 K 0 3 0
13/00	3 5 3	11/20	1 0 2 D 5 K 0 3 3
H 0 4 L 12/56		13/00	3 0 3 B 5 K 0 3 4
審査請求 未請求 請求項の数11 O L (全 9 頁) 最終頁に続く			

(21) 出願番号 特願2000-270778(P2000-270778)

(22) 出願日 平成12年9月6日(2000.9.6)

(71) 出願人 000004226

日本電信電話株式会社

東京都千代田区大手町二丁目3番1号

(72) 発明者 廣津 登志夫

東京都千代田区大手町二丁目3番1号

日本電信電話株式会社内

(72) 発明者 福田 健介

東京都千代田区大手町二丁目3番1号

日本電信電話株式会社内

(74) 代理人 100070150

弁理士 伊東 忠彦

最終頁に続く

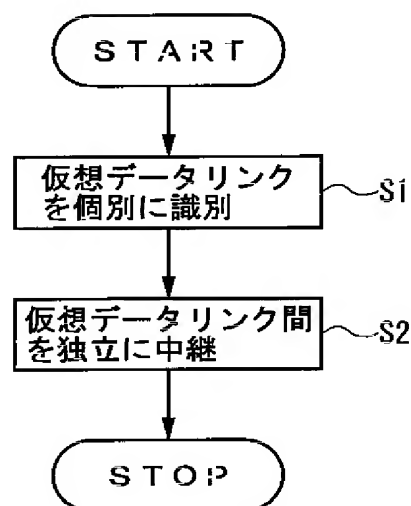
(54) 【発明の名称】 ネットワークによる通信方法及び装置

(57) 【要約】

【課題】 RS毎の独立したプロトコル処理を実現し、独立した複数の仮想ネットワークを実現することが可能な仮想ネットワークによる通信方法及び装置を提供する。

【解決手段】 本発明は、OSやサービスレイヤが物理データリンク上に構成される複数の仮想データリンクを個別に認識し、仮想データリンク間を独立に中継する。

本発明の原理を説明するための図



## 【特許請求の範囲】

【請求項1】 インターネットやプライベートネットワークを介して接続されたネットワークによる通信方法において、  
OS（オペレーティングシステム）やサービスレイヤが物理データリンク上に構成される複数の仮想データリンクを個別に認識し、  
前記仮想データリンク間を独立に中継することを特徴とするネットワークによる通信方法。

【請求項2】 前記仮想データリンク間を独立に中継する際に、  
ネットワーク層やトランスポート層の上位層の情報により通信を区別し、定められた仮想データリンクに中継する請求項1記載のネットワークによる通信方法。

【請求項3】 前記仮想ネットワークと、装置内部の資源とをグループ化して、該装置内の資源を仮想ネットワークに独立に割り付ける請求項1記載のネットワークによる通信方法。

【請求項4】 インターネットやプライベートネットワークを介して接続されたネットワークによる通信方法において、  
複数の仮想データリンクや物理データリンクを関連付けてグループ化したOS内の処理モジュールをリソーススペース（以下、RS）とし、  
前記RS毎に独立したプロトコル処理を行うことを特徴とするネットワークによる通信方法。

【請求項5】 通信の発信元や送信元のIPアドレスやポート番号を含む上位層の情報に応じて、通信を前記RSに割り付けることにより、該上位層の情報に応じて通信を適当な仮想ネットワークに割り付ける請求項4記載のネットワークによる通信方法。

【請求項6】 インターネットやプライベートネットワークを介して接続されたネットワークによる通信装置であって、  
OS（オペレーティングシステム）やサービスレイヤで実現する物理データリンク上に構成される複数の仮想データリンクを個別に認識する認識手段と、前記仮想データリンク間を独立に中継する中継手段とを有することを特徴とするネットワークによる通信装置。

【請求項7】 前記中継手段は、  
ネットワーク層やトランスポート層の上位層の情報により通信を区別し、定められた仮想データリンクに中継する手段を含む請求項6記載のネットワークによる通信装置。

【請求項8】 前記仮想ネットワークと、装置内部の資源とをグループ化して、該装置内の資源を仮想ネットワークに独立に割り付ける手段を有する請求項6記載のネットワークによる通信装置。

【請求項9】 インターネットやプライベートネットワ

ークを介して接続されたネットワークによる通信装置であって、

複数の仮想データリンクや物理データリンクを関連付けてグループ化したOS内の処理モジュールであるリソーススペース（以下、RS）を有し、

前記RS毎に独立したプロトコル処理を行うことを特徴とするネットワークによる通信装置。

【請求項10】 通信の発信元や送信元のIPアドレスやポート番号を含む上位層の情報に応じて、通信を前記RSに割り付けることにより、該上位層の情報に応じて通信を適当な仮想ネットワークに割り付ける手段を有する請求項9記載のネットワークによる通信装置。

【請求項11】 前記RSは、  
前記RS毎のポインタの集合であるポインタ表と、  
アプリケーションのプロセスに付与されている前記RS毎に一意に付与されるRS識別子（RSID）に基づいて、前記ポインタ表を検索して呼び出されるシステムコール処理ルーチンと、  
前記システムコール処理ルーチンの属するRSに基づいて特定され、該RSから特定される仮想インタフェースを介してパケットを処理し、処理したパケットをネットワークに出力するプロトコル処理ルーチンとを有する請求項9記載のネットワークによる通信装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、ネットワークによる通信方法及び装置に係り、特に、単独の計算機、あるいは、ネットワークで接続された複数台からなる計算機、あるいは、インターネットに接続された計算機で構成されたシステムにおけるネットワークによる通信方法及び装置に関する。

## 【0002】

【従来の技術】データリンク層においては、Ethernet（登録商標）（イーサネット（登録商標））やATMなどのように、一つの物理媒体の上に仮想的なデータリンクを複数構成する技術が実現されている。

【0003】Ethernetでは、IEEE802.1Q規格により、伝送するEthernetのフレームに仮想データリンク（仮想LAN：VLAN）の識別子となるタグという数字を保持させ、一つの物理媒体上に仮想的に複数のEthernetのネットワークを構成できる。Ethernetフレームを中継するEthernetスイッチでは、スイッチの各ポートに割り当てられているVLANid（VLAN識別子）に応じて、送られてきたEthernetフレームを転送するかどうかを決定する。これにより、スイッチに接続された機器は所属するVLAN以外のフレームを受信することはない。

【0004】ATMでは、「Virtual Path (VP)」や「Virtual Connection (VC)」という概念で、機器の対の通信接続を、一つの物理媒体上に複数実現することができる。ネットワーク層やトランスポート層では、IP (Int

ernet Protocol) のIPアドレスや、TCP (Transport Control Protocol) や、UDP (User Datagram Protocol) のポート番号をチェックして、ネットワークの一部分に帰着する通信の中継を制限したり、ネットワークの一部分に帰着する通信以外すべての中継を制限したり、また、一部の通信の帯域を制御して、通信品質を変更することができる。これは、ルータのソフトウェアや、ホストOS上の「ipfirewall」、「dummynet」のようなシステムで実現されている。アプリケーション層でも、IPのIPアドレスやTCPやUDPのポート番号によって、ネットワークの一部分に帰着する通信を制限したりすることができる。これは、「TCPwrapper」のようなソフトウェアで実現されている。x-kernel (N.C Hutchins and L.L Peterson, "The x-kernel: An Architecture for Implementing Network Protocols", IEEE Transaction on Software Engineering, 17(1): 64-76, 1991) や、Scout (D Mosberger and L.L. Peterson, "Making Paths Explicit in the Scout Operating System", In proceeding of OSDI '96, 1996) といったOSでは、OS内の処理のモジュール化は行われている。また、Eclipse (J.bruno, E. Gabber, B. Ozden, and A. Silberschatz, "TheEclipse Operating System: Providing Quality of Service via Reservation Domains", In Proceedings of the USENIX 1998 Annual Technical Conference, New Orleans, Louisiana, June 1998) においては、「Reservation Space」(RS) という概念で、OS内部の計算資源の実時間性の確保が実現されている。

【0005】

【発明が解決しようとする課題】しかしながら、仮想データリンクは、ルータやホストのプロトコル処理ソフトウェアでは、通常の物理データリンクと同様に扱われるので、従来のプロトコル処理ソフトウェアでは、ネットワーク層の処理ソフトウェアから、区別無く全ての仮想データリンク及び物理データリンクにアクセスすることができる。従って、仮想データリンク毎に異なる特性（通信品質やセキュリティレベル）を用意して、上位層のプロトコルやアプリケーションからそれらのデータリンクを区別して利用することはできない。

【0006】また、上位層での通信制限は、IPアドレスやポート番号により、仮想的にネットワークを構成していると考えることができ、仮想ネットワークに帰着する通信を区別して使用する。しかし、これらの技術では、通信の起点・終点の情報に応じて通信の制限をするだけで、それらに対してルータやホスト上の計算資源が独立して割り付けられているわけではない。

【0007】「x-kernel」、「Scout」、「Eclipse」といったOSが実現しているのは、OSの処理のモジュール化やそのグループ化であるが、これらは仮想ネットワークと連携しておらず、ネットワークは共有している。また、処理モジュールが概念的に仮想ネットワーク

毎に分離しているわけではない。

【0008】本発明は、上記の点に鑑みなされたもので、ホスト、ルータやスイッチなどのネットワーク機器において、複数の仮想データリンクや物理データリンクを関連付けてグループ化し、そのグループ内の仮想データリンク及び物理データリンクに使用する「処理プロトコル」、「計算資源」、「アプリケーション」を割り付けてグループ化する。この「データリンク」と「処理プロトコル」、「計算資源」、「アプリケーション」のグループを「Resource Space」(リソーススペース: RS) と呼ぶ。このRS毎の独立したプロトコル処理を実現し、独立した複数の仮想ネットワークを実現することが可能なネットワークによる通信方法及び装置を提供することを目的とする。

【0009】また、本発明の目的は、通信の発信元や送信先のIPアドレスやポート番号といった上位層の情報に応じて通信をRSに割り付けることにより、上位層の情報に応じて通信を適当な仮想ネットワークに割り付けることが可能なネットワークによる通信方法及び装置を提供することである。

【0010】

【課題を解決するための手段】図1は、本発明の原理を説明するための図である。

【0011】本発明（請求項1）は、インターネットやプライベートネットワークを介して接続されたネットワークによる通信方法において、OSやサービスレイヤで実現する物理データリンク上に構成される複数の仮想データリンクを個別に認識し（ステップ1）、仮想データリンク間を独立に中継する（ステップ2）。

【0012】本発明（請求項2）は、仮想データリンク間を独立に中継する際に、ネットワーク層やトランスポート層の上位層の情報により通信を区別し、定められた仮想データリンクに中継する。

【0013】本発明（請求項3）は、仮想ネットワークと、装置内部の資源とをグループ化して、該装置内の資源を仮想ネットワークに独立に割り付ける。

【0014】本発明（請求項4）は、インターネットやプライベートネットワークを介して接続されたネットワークによる通信方法において、複数の仮想データリンクや物理データリンクを関連付けてグループ化したOS内の処理モジュールをRSとし、RS毎に独立したプロトコル処理を行う。

【0015】本発明（請求項5）は、通信の発信元や送信元のIPアドレスやポート番号を含む上位層の情報に応じて、通信をRSに割り付けることにより、該上位層の情報に応じて通信を適当な仮想ネットワークに割り付ける。

【0016】本発明（請求項6）は、インターネットやプライベートネットワークを介して接続されたネットワークによる通信装置であって、OSやサービスレイヤで

実現する物理データリンク上に構成される複数の仮想データリンクを個別に認識する認識手段400と、仮想データリンク間を独立に中継する中継手段300とを有する。

【0017】本発明(請求項7)は、中継手段300において、ネットワーク層やトランスポート層の上位層の情報により通信を区別し、定められた仮想データリンクに中継する手段を含む。

【0018】本発明(請求項8)は、仮想ネットワークと、装置内部の資源とをグループ化して、該装置内の資源を仮想ネットワークに独立に割り付ける手段を有する。

【0019】本発明(請求項9)は、インターネットやプライベートネットワークを介して接続されたネットワークによる通信装置であって、複数の仮想データリンクや物理データリンクを関連付けてグループ化したOS内の処理モジュールであるRSを有し、RS毎に独立したプロトコル処理を行う。

【0020】本発明(請求項10)は、通信の発信元や送信元のIPアドレスやポート番号を含む上位層の情報に応じて、通信をRSに割り付けることにより、該上位層の情報に応じて通信を適当な仮想ネットワークに割り付ける手段を有する。

【0021】本発明(請求項11)は、RSにおいて、RS毎のポイントの集合であるポイント表と、アプリケーションのプロセスに付与されているRS毎に一意に付与されるRS識別子(RSID)に基づいて、ポイント表を検索して呼び出されるシステムコール処理ルーチンと、システムコール処理ルーチンの属するRSに基づいて特定され、該RSから特定される仮想インタフェースを介してパケットを処理し、処理したパケットをネットワークに出力するプロトコル処理ルーチンとを有する。

【0022】上記のように、本発明は、一つの物理ネットワーク上に仮想的に複数のネットワークを構成し、各々の仮想ネットワークにそれぞれ異なるセキュリティレベルや通信品質を割り当てることにより、ホストやルータ、スイッチ等で複数のセキュリティレベルや通信品質を利用することが可能となる。

【0023】また、スイッチ等の機器で接続された機器に対する仮想ネットワークの割当を制限することにより、ホストやネットワークに対して特定のセキュリティレベルや通信品質の利用を制限することが可能である。

【0024】また、本発明により、“Ethernet VLAN”や“ATM VC”のような仮想データリンクを利用していないネットワークを流れる通信を上記で実現した様々なセキュリティレベルや通信品質の仮想ネットワークに振り分けることが可能となる。さらに、本発明では、上記で実現したセキュリティレベルや通信品質の仮想ネットワークに応じて、独立したプロトコル処理やアプリケーションの実行が可能になる。これにより、例えば、特定の

通信品質の仮想ネットワークのみで稼働するアプリケーションを実現したり、特定のセキュリティレベルではIPレベルで中継し、他のセキュリティレベルでは、アプリケーション層で中継したりといった細かい制御を実現したりすることが可能になる。

【0025】

【発明の実施の形態】まず、以下の説明で用いる、仮想データリンク、仮想ネットワーク、仮想インタフェースの用語を定義する。

【0026】仮想データリンク(Virtual Datalink: VDL)は、単一の物理ネットワークセグメント内での一つの通信クラスであり、1つのVDLは、それを識別する何らかの識別子が必要になる。識別子の候補としては、次のようなものが考えられる。

【0027】・802.1Q VLAN tag: この場合、一つのVLAN idを割り当てられた一セグメントが一つのVDLとなる。

【0028】・IPのネットワークアドレスとネットマスクの組: この場合、物理インタフェースには複数のIPアドレスが付与され、一つのアドレスが一つのVDLを表す。

【0029】・source port (ソースポート) 番号: この場合、source port 番号の特定の範囲が一つのVDLを表す。TCPのセッションを開く時に、VDLに応じて適当なsource port を付けたり、UDPのパケット送付の際に適当なsource port を付ける。仮想ネットワーク(Virtual Network: VNET)は、複数のVDLを接続して構築される仮想的なネットワークであり、接続する2つのVDLで、その識別子が同じである必要はない。仮想ネットワークは、ルータやゲートウェイを越えて複数のデータリンクに跨がるネットワークの概念である。単一の物理ネットワーク上には複数の仮想ネットワークが存在し得る。

【0030】仮想インタフェース(Virtual Datalink Interface: VDI)は、一つの仮想ネットワーク(VNET)へのOS内部でのインタフェースである。

【0031】次に、仮想ネットワークを図面を用いて説明する。

【0032】図3は、本発明の仮想ネットワークを説明するための図である。

【0033】同図では、組織内ネットワーク上において、3つの仮想ネットワーク(vnet0, vnet1, vnet2)が重なっている。各物理データリンク上では、仮想ネットワーク(VET)が仮想データリンク(VDL)として区別されており、区別されたままルータなどで中継されることで複数の物理データリンクにわたる仮想ネットワーク(VNET)を構成している。

【0034】このように一つの物理的なネットワークの上に、複数の仮想ネットワークが重なって存在するように見え、多重化された仮想ネットワークの空間を構成す

る。仮想データリンクとして802.1Q VLANを使う場合を考えると、仮想ネットワーク(VNET)中全域で、VLANtag id (VLANタグ識別子)が一意である必要はない。中継するルータの両側で異なるtag idを用いていても構わない。ここで必要なのは、同一仮想ネットワーク(VNET)を構成する別物理データリンクの仮想データリンク(VDL)間で、tag idの対応付けが正しく行われることである。つまり、例えば、vnet0を構成するvdl00とvdl10について、vdl00のtag idが10で、vdl10のtag idが25であっても構わないということである。

【0035】同図の仮想ネットワークでは、インターネットへ接続しているルータGは、vnet0にしかアクセスできない。ルータBは、通常のファイアウォールを構成するフィルタリングルータで、インターネットから内部への直接のアクセスを排除するために、vnet0の外部から通信を遮断している。このファイアウォールセグメント上に外部からのアクセスのために用意されたゲートウェイAでは、アプリケーションゲートウェイがvent0からvnet2への中継を行っている。これにより、以後内部ネットワークにアクセスする際には、vnet2を通してアクセスすることになり、例えば、遠隔ログインを繰り返しても、内部ネットワークではその通信の起点が外部にあるものと判断できる。

【0036】次に、ホストやルータ内部でのOS(Operating System)の処理について説明する。

【0037】図4は、本発明のOS内部の処理を説明するための図である。

【0038】各仮想ネットワーク(VNET)に応じてホストやルータ内部での処理を変更するために、ホストやルータ内部のOSにおいて、独立な処理モジュールのグループと、それらの複数モジュールグループ間の切替機構が必要になる。OS内の処理モジュールのグループをResource Space(RS)と呼ぶ。RSの実体は、処理モジュール中の関数及び、処理モジュール毎のローカルな状態を保持するデータを集めたもので、各々のRSは、識別子(RSID)を持つ。この処理モジュールとは、IPやTCPのプロトコル処理プログラムや、exec()、open()といったシステムコールを受けてOS内部で実際の処理をするシステムコール関数の処理プログラムである。

【0039】各仮想インタフェース(VDI)は、同図中、パケット分別器(packet classifier)で分別されることにより具現化される(同図中、vent0やvnet1の楕円)。IPアドレスやポート番号といった上位層の情報に応じた仮想ネットワークへの振り分けは、分別の対象として上位層の情報を使うことにより、このパケット分別器の分別により実現される。

【0040】各仮想インタフェース(VDI)には、対応するRSIDが決まっており、受け取ったパケットに

対してRSIDを付属情報として張り付けてOS内部での処理を進める。

【0041】アプリケーション(同図中一番上の楕円)側から見ると、各プロセスにRSIDが付いており、システムコールを発呼してOS内部に処理が移った段階で、RSIDに応じた処理プログラムが呼び出される。

【0042】これにより、例えば、“exec()”に対して、“setuid”できない“exec処理ルーチン”を用意しておいて、通信クラスに応じて“setuid”を禁止することなどが可能になる。

【0043】次に、OS内部の詳細な構造について説明する。

【0044】図5は、本発明の多重通信クラス実装時のOS内部の構成を示す図である。同図において、右半分の本構造は、プロトコル処理階層及びシステムコールの概念的な構造を表している。同図中左半分の斜字体の各モジュールは、プロトコル処理モジュール(pIP0, pTCP0)やプロトコル依存データ(dIP0, dTCP0)、システムコール処理ルーチン(open0, exec0など)といった処理実体の構造を表している。

【0045】各RSは、これらのモジュールの集合として実現されるので、RSの実体は、プロトコル処理モジュール/データ、システムコール処理ルーチンへのポインタの集合(同図中、RSnのポインタ表)となる。

【0046】プロトコル処理モジュールについては、プロトコル依存のデータ(例えば、IPモジュールでは、IP forwardingするかどうかの変数など)を必要に応じてRS毎に独立にすることにより、同一のプロトコル処理モジュールを使いながら、RSにより処理を変えることも可能である。例えば、同一のIP処理モジュールpIP0を用いて、あるVNETではIP forwardingを許すが、他のVNETでは許さないといったことの実現が可能である。

【0047】ネットワークから入ってきたパケットは、パケット分別器で分別され、仮想インタフェース(VDI:vnet0等)に渡される。VDIは、RSに対応付けられているので、ここでRSのポインタ表へのポインタがパケットの管理情報として付加されて、以後の処理に渡される。RSが決まれば、処理モジュールの表(RSのポインタ表)が確定するので、以後の処理は、RSの関数とデータを使って進められる。一方、ネットワークにパケットを送出する場合には、プロセスに付加されているRSIDを使って進められる。一方、ネットワークにパケットを送出する場合には、プロセスに付加されているRSIDをもとにRSのポインタ表を見つけて、RS毎に処理の切替を実現する。

【0048】

【実施例】以下、図面と共に本発明の実施例を説明する。

【0049】OS内部の処理について説明する。

【0050】図6は、本発明の一実施例の通信装置の構成を示す。

【0051】同図に示す装置は、アプリケーション100、システムコール200、OS300、及びパケット分別器400から構成される。

【0052】アプリケーション100は、プロセス101、102、103、104を有する。

【0053】OS300は、RS310、320を有し、それぞれのRS310、320は、プロトコル・処理ルーチン311、321とシステムコール・処理ルーチン312、322を有する。なお、同図では、説明の簡化のため、RS310、320の2つのRSのみ示してあるが、n個のRSが存在する。

【0054】パケット分別器(packet classifier)400は、ネットワークからのパケットの入力を受け、各プロトコル階層のヘッダ情報などを用いて、適切な仮想インタフェース(同図中、vnet0、vnet1)に分別する。分別のための規則は、ルール表410の形で当該パケット分別器400が保持している。このルール表410は、OSの構成時に決定され、システムコール200を通じてアプリケーション100からの制御で変更が可能である。特定のRS310、320中のシステムコール処理ルーチン312、322を、このルール表410が変更できないものにしておくことで(同図中のRS320のpktctl'ルーチン)、ルール表410の制御が不可能なRSを作ることが可能であり、このRS320上で稼働するアプリケーション100(プロセス)についてルール表410の変更を禁止することができる。

【0055】パケット分別器400で仮想インタフェースに分別されたパケット(vnet0,vnet1)は、その仮想インタフェースに対応するRS310、320のプロトコル処理ルーチン(TCP,UDP,IP,rt等)311、321に渡される。プロトコル処理ルーチン311、321は、IP,TCP,UDP等のプロトコルの実際の処理をするルーチンである。このプロトコル処理ルーチン311、321は、RS310、320毎に保持するポインタ表313、323で管理されており、プロトコル処理ルーチン間の関係もこの表で決定する。

【0056】アプリケーション100からの処理要求は、システムコール200を介して処理される。システムコール200の引数として、RSIDが渡される場合は、システムコール処理ルーチン(open,exec等)312、322が呼び出される前に、そのプロセスが指定されたRSの処理を使ってよいかどうかの許可表330を検索し、権限を調べる。RSIDが渡されない場合は、プロセスに付与されているRSIDを用いる。システムコール処理ルーチン312、322は、RSIDをもってRS対応表340検索し、該当するRSのポインタ表313、323を獲得することで、当該ポインタ表313、323に指し示されているシステムコール

処理ルーチン(open,exec等)312、322を呼び出す。

【0057】以下に、上記の構成における処理を説明する。

【0058】アプリケーション100からネットワークやディスク等の周辺機器に対する読み書きの処理は、システムコール200を通じて行われる。システムコールは、アプリケーションとOS内部の処理の切り替えを行うインタフェースである。

【0059】図7は、本発明の一実施例のシステムコールの処理を示すフローチャートである。同図に示す処理は、ソケットやファイルのオープン、読出し要求、書込み要求など全てのシステムコールの処理に共通する。

【0060】まず、アプリケーション100がシステムコール200を呼び出す(ステップ101)。次に、RSIDが引数に指定されているかを確認し(ステップ102)、指定されていない場合には、アプリケーション100のプロセスに付与されているRSIDを調べ(ステップ103)、RSIDから該当のRSのポインタ表(313、323)を特定し、システムコール処理ルーチン(312、322)を呼び出す(ステップ105)。指定されている場合には、許可表330でそのプロセスに許可されているRSかを判定し(ステップ104)、RSIDから該当のRSのポインタ表(313、323)を特定し、システムコール処理ルーチン(312、322)を呼び出す(ステップ105)。

【0061】次に、ネットワークから入力されたパケットの処理について説明する。

【0062】図8は、本発明の一実施例のネットワークからの入力処理のフローチャートである。

【0063】ネットワークからパケット分別器400にパケットが到着すると(ステップ201)、データリンク層、ネットワーク層、トランスポート層などのヘッダを見て、ルール表410を検索する(ステップ202)。検索した結果該当するものがあれば(ステップ203、Yes)、該当したルールに対する仮想インタフェースを用い(ステップ204)、該当するものがなければ(ステップ203、No)、デフォルトの仮想インタフェースを用いる(ステップ205)。次に、仮想インタフェースを通じてRSのポインタ表(313、323)を設定し、プロトコル処理ルーチン(311、321)にパケットデータを渡す(ステップ206)。プロトコル処理ルーチン(311、321)は、RSのプロトコルスタックに応じてパケットを処理し(ステップ207)、アプリケーション100の読出し(システムコール)を待つ(ステップ208)。

【0064】上記の流れで処理されたデータは、読出しのシステムコール200を通じてアプリケーション100に渡される。

【0065】次に、アプリケーション100からネット

ワークに書き出されたデータの処理について説明する。

【0066】図9は、本発明の一実施例のネットワークへの出力処理のフローチャートである。

【0067】システムコール200などを通じてデータが書き込まれ(ステップ301)、システムコール処理ルーチン(312、322)の属するRS310、320から、プロトコル処理ルーチン(311、321)が特定されると(ステップ302)、プロトコル処理ルーチン(311、321)でデータを処理し(ステップ303)、RS310、320から特定される仮想インタフェースを通じてデータ(パケット)をネットワークに送出する(ステップ304)。

【0068】上記の処理において、データはシステムコール200を通じてOS300内の処理ルーチンに渡され、プロトコル処理を通してネットワークに送出される。

【0069】上述のように、本発明では、OSやサービスレイヤで、通信トポロジを管理すると共に、仮想データリンクを個別認識し、独立に中継することにより、通信の起点に応じてアクセス制御を行うことができ、ネットワークのセキュリティを容易に高めることが可能となる。

【0070】なお、本発明は、上記の実施例に限定されことなく、特許請求の範囲内において種々変更・応用が可能である。

【0071】

【発明の効果】上述のように、本発明によれば、一つの物理ネットワーク上にセキュリティレベルや通信品質の異なる複数の仮想的なネットワークを構成し、それらの仮想ネットワーク毎に適切なプロトコル処理やアプリケーション実行が可能になる。これにより、例えば、インターネットに晒された比較的危険なネットワークや、社内の多数のユーザがアクセスするネットワーク、部署内のユーザだけがアクセスするネットワーク、一人のユーザだけがアクセスするネットワークといった複数のセキュリティレベルに応じた仮想ネットワークを用意し、それぞれに稼働させるアプリケーションを制限したり、通信の中継方法をIP層の中継やアプリケーション層の中

継など多様に使い分けることにより、ネットワークに対するセキュリティをきめ細かに制御することが可能になる。

【0072】また、これらの仮想ネットワークの提供をスイッチ等のネットワーク機器側から制御することも可能となる。これにより、初心者から上級者に至るまでの様々な技術のユーザに応じた、適切なインターネットアクセス及びイントラネットアクセスを提供することができる。

【図面の簡単な説明】

【図1】本発明の原理を説明するための図である。

【図2】本発明の原理構成図である。

【図3】本発明の仮想ネットワークを説明するための図である。

【図4】本発明のOS内部の処理を説明するための図である。

【図5】本発明の多重通信実装時のOS内部の構成図である。

【図6】本発明の一実施例の通信装置の構成図である。

【図7】本発明の一実施例のシステムコールの処理を示すフローチャートである。

【図8】本発明の一実施例のネットワークからの入力処理のフローチャートである。

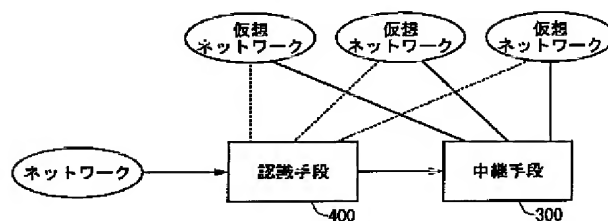
【図9】本発明の一実施例のネットワークへの出力処理のフローチャートである。

【符号の説明】

100 アプリケーション  
101、102、103、104 プロセス  
200 システムコール  
300 OS、中継手段  
310、320 RS (Resource Space)  
311、321 プロトコル処理ルーチン  
312、322 システムコール処理ルーチン  
330 許可表  
340 RS対応表  
400 認識手段、パケット分別器  
410 ルール表

【図2】

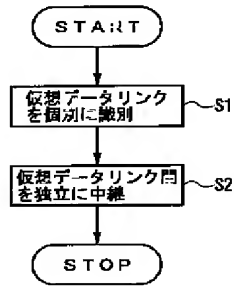
本発明の原理構成図





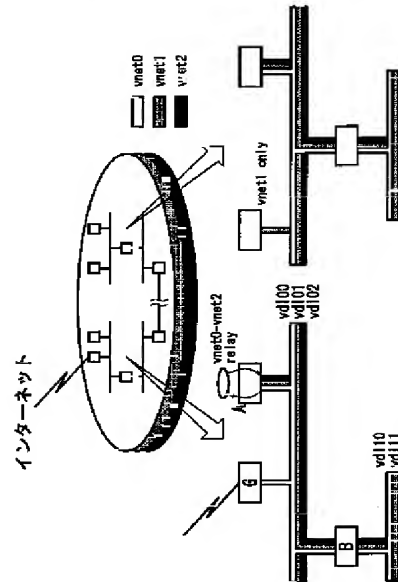
【図1】

本発明の原理を説明するための図



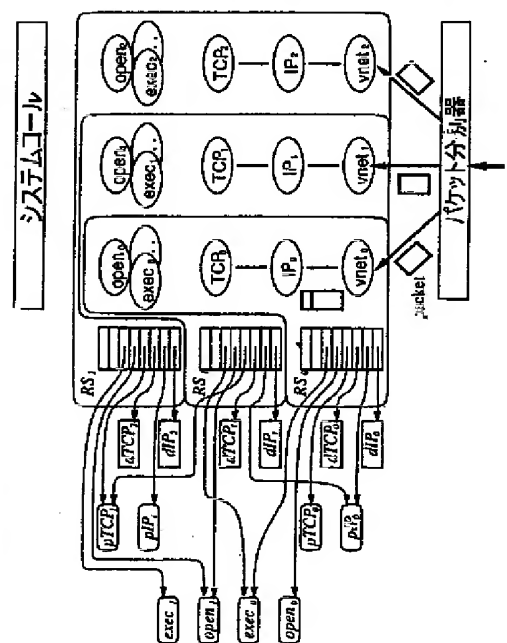
【図3】

本発明の仮想ネットワークを説明するための図



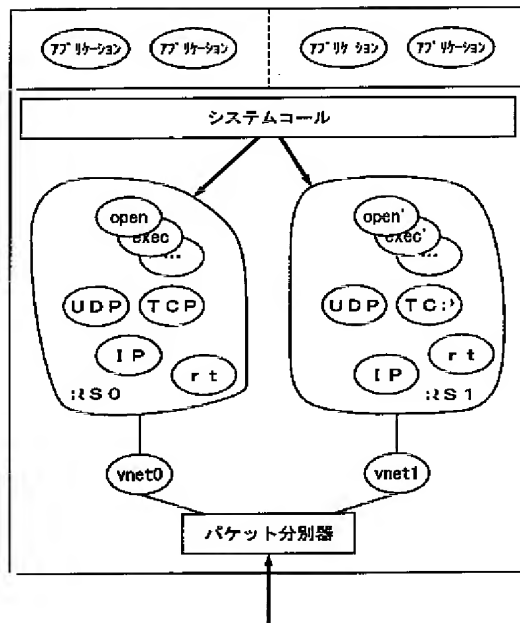
【図5】

本発明の多重通信実装時のOS内部の構成図



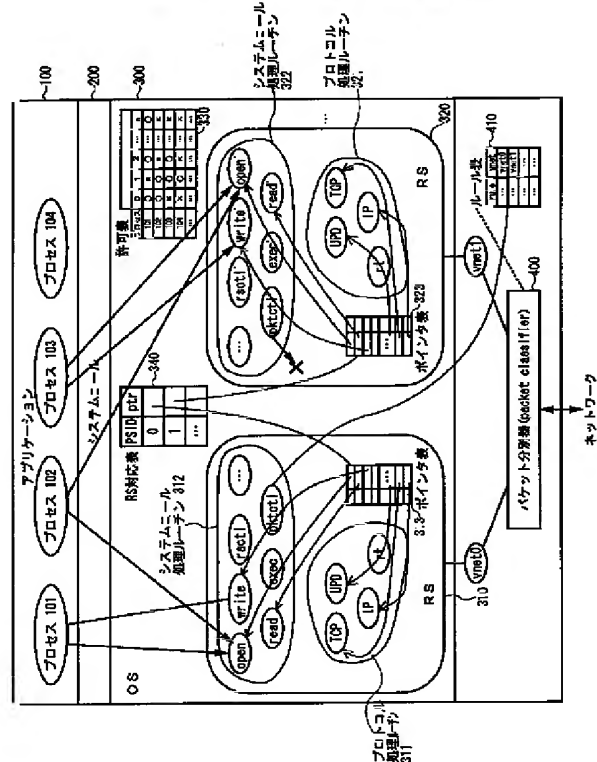
【図4】

本発明のOS内部の処理を説明するための図



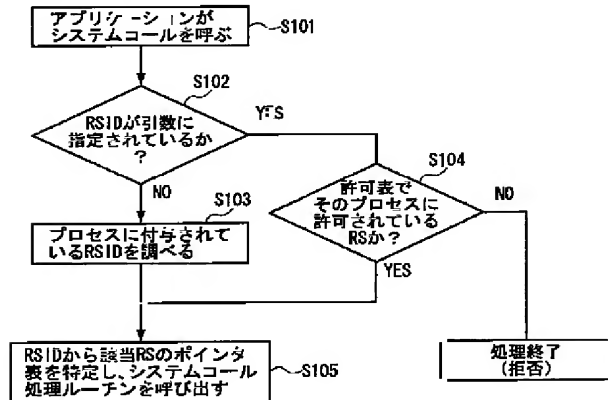
【図6】

本発明の一実施例の通信装置の構成図



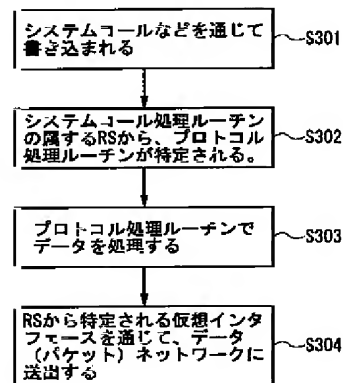
【図7】

本発明の一実施例のシステムコールの処理を示すフローチャート



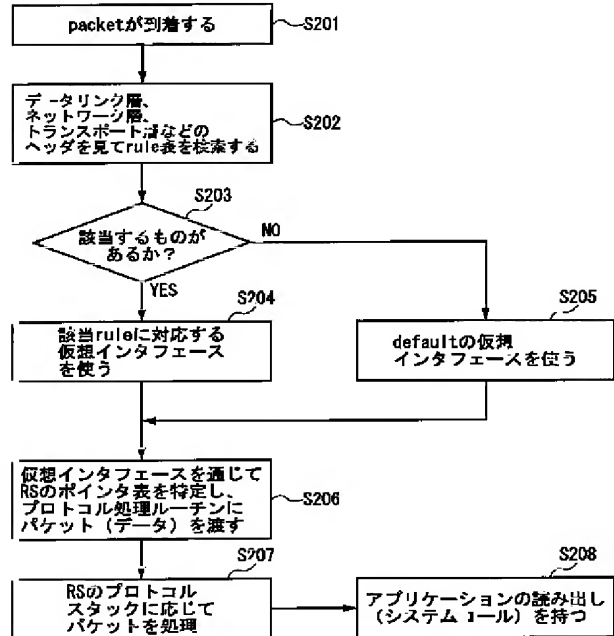
【図9】

本発明の一実施例のネットワークへの出力処理のフローチャート



【図8】

本発明の一実施例のネットワークからの入力処理のフローチャート



フロントページの続き

(51) Int. Cl.<sup>7</sup>

H 0 4 L 29/04

識別記号

F I

(参考)

- (72) 発明者 明石 修  
東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内
- (72) 発明者 佐藤 孝治  
東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内
- (72) 発明者 山崎 憲一  
東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内

- (72) 発明者 菅原 俊治  
東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内
- F ターム(参考) 5B089 GA04 GB01 HB19 KA04 KB03  
KB06 KF06 KG05 KG08  
5B098 GA02 GA04 GC01  
5K030 HC01 HC13 HD03 LB05  
5K033 BA04 CB08 CC01 DA05 DB18  
5K034 JJ24 KK21 KK27 KK28